

*NASA Contractor Report 201625*

*ICASE Report No. 96-68*

# ICASE

## ON THE OPTIMAL NUMBER OF SUBDOMAINS FOR HYPERBOLIC PROBLEMS ON PARALLEL COMPUTERS

**Paul Fischer**  
**David Gottlieb**

*NASA Contract No. NAS1-19480*  
*December 1996*

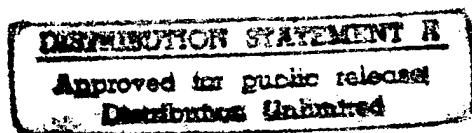
*Institute for Computer Applications in Science and Engineering*  
*NASA Langley Research Center*  
*Hampton, VA 23681-0001*

*Operated by Universities Space Research Association*



*National Aeronautics and*  
*Space Administration*

*Langley Research Center*  
*Hampton, Virginia 23681-0001*



19970214 019

DTIC QUALITY INSPECTED 1

# ON THE OPTIMAL NUMBER OF SUBDOMAINS FOR HYPERBOLIC PROBLEMS ON PARALLEL COMPUTERS

Paul Fischer <sup>1 2</sup> and David Gottlieb <sup>2 3</sup>

Division of Applied Mathematics

Brown University

Providence, RI 02912

## Abstract

The computational complexity for parallel implementation of multidomain spectral methods is studied to derive the optimal number of subdomains,  $q$ , and spectral order,  $n$ , for numerical solution of hyperbolic problems. The complexity analysis is based upon theoretical results which predict error as a function of  $(q, n)$  for problems having wave-like solutions. These are combined with a linear communication cost model to study the impact of communication overhead and imposed granularity on the optimal choice of  $(q, n)$  as a function of the number of processors. It is shown that, for present day multicomputers, the impact of communication overhead does not significantly shift  $(q, n)$  from the optimal uni-processor values, and that the effects of granularity are more important.

---

<sup>1</sup>Supported by NSF Grant ASC-9405403.

<sup>2</sup>Supported by AFOSR Grant F49620-95-1-0074.

<sup>3</sup>This research was partially supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the second author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

# 1 Introduction

We examine the computational efficiency for parallel solution of a hyperbolic partial differential equation in  $d$ -space dimensions and time on  $\Omega = [-1, 1]^d \times [0, \tau]$ . We consider a class of model problems in which temporal discretization is assumed to be based upon an explicit marching procedure and spatial discretization is based upon a spectral subdomain approach with  $Q$  subdomains, each having order of approximation  $n$  in each spatial direction. The choice of  $Q$  and  $n$  is determined by the dual constraints that the discretization error must be below a certain level, and that the computational time should be minimized. This optimization problem has been previously analyzed for the uni-processor case [1]. Here, we consider the impact of non-uniform memory access times imposed by parallel distributed memory architectures.

Examples of architecture and discretization dependent computational cost analyses have also been presented in [2, 3, 4, 5]. Keller and Schreiber [2] study the costs for parameter continuation of steady-state Navier-Stokes calculations on vector supercomputers with a cost-functional that incorporates both CPU and memory charges, and with the principal independent parameter being the frequency of full-Newton vs. chord continuation steps. Chan and Shao [3] study the question of the optimal number of subdomains for the solution of elliptic problems via domain decomposition based iterative solvers. In that study, a subdomain defines the support of local block preconditioners and corresponding coarse-grid operators, rather than redefining the underlying discretization. Rønquist [4] examines the trade-off between number of subdomains and local approximation order for the spectral element method applied to serial solution of model elliptic problems. Fischer and Patera [5] provide both theoretical and experimental performance analysis for the spectral element method on hypercubes.

In the present study, we use an explicit error estimate for hyperbolic problems having wave-like solutions as derived by Gottlieb and Wasberg [1] to determine the optimal discretization for a tensor-product based model problem on distributed memory architectures. Although fairly simple, this model embodies many essential discretization/architectural details which must be considered in designing an algorithm for modern day supercomputers. These include: the trade off between high- and low-order accuracy discretizations, with compensating resolution to maintain a fixed accuracy; the cost of non-local memory accesses, accounted for by a message-passing model; and the imposition of a fixed granularity due to the fact that  $P$  processors are employed in the simulation. It should be noted that, in contrast to high-order finite difference schemes, the spectral subdomain approach is in fact a heterogeneous discretization in the sense that partitioning the domain along subdomain boundaries induces far less communication than would arise if the subdomains themselves were subdivided. This heterogeneity is well suited to computer architectures exhibiting a two-tiered (local, and non-local) memory access cost.

## 2 A Hyperbolic Model Problem

As a model problem, we consider the  $d$ -dimensional convection equation on  $\Omega = [-1, 1]^d \times [0, \tau]$  given by:

$$\begin{aligned} \frac{\partial \phi}{\partial t} + \nabla \cdot (\vec{U} \phi) &= f(\vec{x}, t) \\ \phi(\vec{x}, 0) &= \phi_0(\vec{x}), \end{aligned} \quad (1)$$

where the unknown,  $\phi$ , is a scalar convected with a given velocity field,  $\vec{U}(\vec{x}, t)$ , and  $f$  is a known forcing function. Appropriate boundary conditions are assumed on each face of the domain – the details are not important to the cost analysis considered here. We assume that (1) is discretized in time via an explicit time-marching procedure requiring only local vector updates and no system solves to advance the solution at each step.

Spatial discretization is based upon a spectral subdomain approach. Let  $Q = q^d$  be the total number of subdomains, and  $N = n^d$  be the number of points within each subdomain;  $q$  and  $n$  are the number of subdomains and points (order of approximation) in *each* spatial direction, respectively. The total number of grid points is therefore  $QN = (qn)^d$ . The  $d$ -dimensional domain is constructed as a tensor product of one-dimensional partitions of  $[-1, 1]$ . We assume a uniform partition in each direction, yielding, e.g., a  $q \times q$  array of  $(n \times n)$  squares in two dimensions. Note that in the limit,  $n \rightarrow 1$ , we recover a low-order finite difference or finite element approximation. In the other extreme,  $q \rightarrow 1$ , we recover a spectral discretization.

The discretization leads to an evolution equation of the form:

$$\underline{\phi}^{k+1} = \underline{\phi}^k + \sum_{i=0}^{l-1} \beta_i \left[ \underline{\nabla} \cdot \underline{\vec{F}}^{k-i} + \underline{f}^{k-i} \right]. \quad (2)$$

Here, underscore denotes vectors of basis coefficients,  $\underline{\nabla}$  is the discrete divergence operator,  $\underline{\vec{F}}$  is the vector of fluxes:  $\{\underline{\vec{F}}\}_j = \vec{U}_j \phi_j$ ,  $j = 1, \dots, QN$ , and the  $\beta_i$ 's denote the coefficients for an  $l$ th-order time stepping scheme (e.g. [6]). More stable Runge-Kutta schemes can be accommodated without significantly altering the computational complexity.

We assume that within each subdomain, derivatives are evaluated via application of a one-dimensional derivative matrix in a tensor-product fashion, e.g., in  $\mathbb{R}^3$ ,

$$\frac{\partial F_x}{\partial x} \rightarrow [I_z \otimes I_y \otimes D_x] \underline{F}_x,$$

where  $I_y$  and  $I_z$  are the  $(n \times n)$  identity matrices and  $D_x$  is the  $(n \times n)$  differentiation matrix which may be derived from a collocation method or high-order finite difference method. In spectral subdomain methods the derivatives at the interfaces are, in effect, evaluated with nonsymmetric stencils. To maintain continuity of the solution, information needs to be propagated across subdomain boundaries. Stability and  $n$ th-order accuracy (i.e., exponential with  $n$ ) can be achieved just by propagating surface function values across the interface; higher derivatives or additional stencil values do not need to be shared across the interface [7]. Consequently, the communication cost

is essentially the same as that for a low-order scheme, i.e., proportional to the surface area of the subdomain, and not to the volume.

For the model problem (2) the spatial truncation error is governed by the choice of discretization for the gradient operator. For solutions exhibiting wave-like behavior with maximum wave number  $k$ , the spectral subdomain discretization leads to an error estimate of the form [1]:

$$\text{error} = \left( \frac{e\pi k}{2qn} \right)^n. \quad (3)$$

Two possible convergence strategies ensue. One can increase the number of subdomains, leading to algebraic convergence of order  $n$ , or increase the approximation order within each subdomain, leading to exponential convergence.

Throughout the remainder of the paper, we assume that the error is constrained to a user specified value, i.e., we require

$$\left( \frac{e\pi k}{2qn} \right)^n \leq e^{-\epsilon}, \quad (4)$$

and seek to minimize computational cost (time) subject to the constraint that  $\epsilon$  is fixed. Note that for fixed error  $\epsilon$ , an increase in approximation order ( $n$ ) implies a decrease in the number of subdomains ( $q$ ), and vice-versa. This result derives directly from the error constraint (4) independent of the subsequent computational complexity analysis. Moreover, for  $\epsilon$  fixed, an increase in  $n$  also implies a decrease in the number of grid points,  $QN$ , though not necessarily a reduction in the total work.

### 3 Computational Complexity

We consider the implementation of the model problem (2) on a distributed memory MIMD parallel architecture. The programming model is based upon the single program, multiple data (SPMD) paradigm; each processor is assumed to have its own private address space and non-local memory is accessed via explicit message passing. Letting  $P = p^d$  be the total number of processors to be employed, we assume that the subdomains can be integrally mapped onto the processors in each direction, implying that  $q = mp$ ,  $m$  a positive integer. The discretization on each processor therefore consists of an  $m^d$  array of blocks, with each block containing an  $n$ th-order spectral discretization based, e.g., on tensor-product Chebyshev polynomials of degree  $n$ .

We assume that the communication time is not "covered" by computation, i.e., that arithmetic operations do not take place simultaneously with communication. The  $P$ -processor solution time  $T_P$  is therefore the sum

$$T_P = T_a + T_c, \quad (5)$$

where  $T_a$  and  $T_c$  denote the arithmetic and communication components, respectively. The optimal number of subdomains is determined by minimizing  $T_P$  subject to the error tolerance constraint (4).

Advancement of the solution (2) requires several vector-vector updates, each with an operation count  $O(QN) = O(q^d n^d)$ . In addition, the discrete divergence

operator requires one derivative evaluation in each of  $d$  spatial directions. Because of the tensor product form of the bases, the operation count for each  $n$ th-order derivative evaluation scales as  $O(d q^d n^{d+1})$  for matrix-matrix product based differentiation, and  $O(d q^d n^d \log n)$  for fast-transform approaches. We denote the combined work estimate as:

$$\mathcal{N}_{ops} = A q^d n^{d+\gamma},$$

with  $0 < \gamma < 1$ . Here,  $A(d)$  is a small parameter proportional to  $d^2$  but independent of  $q$  and  $n$ . The non-integer exponent  $\gamma$  provides the flexibility to account for the lower order terms in the operation count and for the fact that, even in the matrix-matrix product case, the *time* for the leading order term will typically scale less rapidly than the  $O(n^{d+1})$  operation count as vector performance generally improves with increasing  $n$ .

Aside from the interface data exchanges, full parallelism is attained for the vector update (2), yielding a per-step arithmetic time complexity of:

$$T_a = A \frac{q^d n^{d+\gamma}}{P} t_a, \quad (6)$$

where  $t_a$  is the (average) time required for a single arithmetic operation such as multiplication or addition. For matrix-matrix product based differentiation,  $t_a$  will be close to the processor clock-cycle time, as good use is made of local memory hierarchies (i.e., cache). For fast-transform approaches,  $t_a$  will be significantly larger. However, for sufficiently large  $n$ , this is clearly balanced by the  $O(n \log n)$  complexity.

In the multi-processor implementation, communication is required at each time-step to update edge values shared by adjacent subdomains. For most modern message passing parallel computers, an appropriate model of contention-free communication is given by the linear function:

$$t_{comm}[w] = (\alpha + \beta w) t_a.$$

Here,  $t_{comm}[w]$  is the time required to transmit an  $w$ -word message from one processor to another,  $\alpha$  and  $\beta$  are non-dimensional constants representing message start-up time (latency) and asymptotic per-word transfer time, respectively. In appropriate units,  $\beta$  is the inverse of the communication bandwidth. For the  $d$ -dimensional model problem, each processor transmits

$$w = B \left( \frac{qn}{p} \right)^{d-1}$$

words per step, where  $d \leq B \leq 2d$  is a parameter which is dependent on the problem and possibly varying from one subdomain to the next. If the local convection is predominantly in one direction, information will flow from only  $d$  faces of each subdomain, rather than from each of the  $2d$  faces present on each subdomain. We will consider the suboptimal case where information must propagate in each direction.

In addition to the face data, there is also  $O(n^{d-1}) \dots O(1)$  edge and vertex data which must be communicated. Although of a lower-order than the principal face data which, these terms cannot be ignored because latency effects keep communication time

from going to zero as the message size decreases. However, we do not directly account for these lower order terms in the present model for the following reasons. First, for a  $d$ -dimensional tensor product geometry such as employed here, it is possible to properly exchange the lower-order values in the course of exchanging the  $2d$  faces *without any extra communication* (see, e.g., [5]). Second, in cases where the mesh topology does not permit such efficiencies, it is unlikely that the choice of the optimal  $(q, n)$  pair will be strongly influenced by the edge exchanges; they generally comprise short messages dominated by latency costs which are independent of  $(q, n)$ . We reconsider this point in the results section.

Denoting the total communication cost as  $T_c$ , we have:

$$T_c = (1 - \delta_{1p})(T_\alpha + T_\beta) ,$$

where the Kronecker delta term,  $(1 - \delta_{1p})$  accounts for the absence of communication in the uni-processor case. Here,

$$T_\alpha = B\alpha t_a \quad (7)$$

is the latency term, and

$$T_\beta = B \frac{(qn)^{d-1}}{p^{d-1}} \beta t_a \quad (8)$$

is the bandwidth component.

Combining (6-8), the  $P$ -processor solution time is:

$$T_P = t_a \left[ A \frac{q^d n^{d+\gamma}}{p^d} + \left( B \frac{(qn)^{d-1}}{p^{d-1}} \beta + B\alpha \right) (1 - \delta_{1p}) \right] . \quad (9)$$

It is clear that, for fixed problem size,  $(q, n)$ , latency becomes the dominant factor in loss of parallel efficiency as  $p \rightarrow \infty$ . However, the latency term has no  $(q, n)$ -dependence and therefore does not influence the optimal discretization choice.

## 4 Cost Analysis

We now consider the problem of finding a discretization pair  $(q, n)$  which minimizes  $T_P$  subject to the error tolerance constraint (4). Clearly, since  $T_\alpha$  is independent of  $q$  and  $n$ , we need only consider the sum of  $T_a$  and  $T_\beta$ .

The dimension of the parameter space  $(q, n)$  can be reduced from two to one by using the constraint (4) to solve for  $qn$  in terms of  $n$ :

$$qn = \frac{e\pi k}{2} e^{\frac{\epsilon}{n}} . \quad (10)$$

Substituting this into (6) and (8):

$$T_a = t_a A \left( \frac{e\pi k e^{\frac{\epsilon}{n}}}{2p} \right)^d n^\gamma \quad (11)$$

$$T_\beta = t_a \beta B \left( \frac{e\pi k e^{\frac{\epsilon}{n}}}{2p} \right)^{d-1} . \quad (12)$$

Differentiating with respect to  $n$  leads to

$$\frac{d}{dn}T_a = \frac{\gamma n - \epsilon d}{n^2}T_a \quad (13)$$

$$\frac{d}{dn}T_\beta = \frac{(1-d)\epsilon}{n^2}T_\beta \quad (14)$$

Equating the sum of (13) and (14) to zero, we find:

$$n_{opt} = \frac{\epsilon}{\gamma} \left[ d + (1 - \delta_{1p})(d-1) \frac{T_\beta}{T_a} \right] \bigg|_{n=n_{opt}}, \quad (15)$$

where the Kronecker delta term has been incorporated to reflect the uni-processor case.

Note that (15) is an implicit equation for  $n_{opt}$ , as  $T_a$  and  $T_\beta$  are dependent on  $n$ . However, for the parameters under consideration, the ratio  $T_\beta/T_a$  is not a strong function of  $n$ , and a fixed-point iteration in the neighborhood of  $n \approx n_{opt}$  typically converges in one or two iterations. Consequently, the structure of (15) reveals much of the expected behavior of  $n_{opt}$ . For example, when  $\beta$  is small or  $d = 1$ , the communication cost is negligible and we recover the optimal order for the serial case,  $n_{opt_s} = \frac{\epsilon d}{\gamma}$ . In this case, the optimal order increases (and correspondingly, the number of subdomains decreases) with increasing spatial dimension,  $d$ , decreasing error tolerance,  $e^{-\epsilon}$ , or decreasing spectral overhead cost,  $\gamma$ . It is interesting to note that under these circumstances, the optimal choice of  $n$  is independent of  $k$ , and from (4) we conclude that the optimal number of subdomains,  $q_{opt}$ , is proportional to the maximum resolvable wave-number,  $k$ .

In the general case, an explicit formula for  $n_{opt}$  can be derived by setting  $z = \frac{n_{opt}}{\epsilon}$ , and substituting into (15). Using (11-12), we can derive an equation in which the  $z$ -dependence is explicit:

$$f(z) \equiv z^\gamma \left( z - \frac{d}{\gamma} \right) e^{\frac{1}{z}} = \frac{d-1}{\gamma} \frac{\beta B}{A} \frac{2p}{e\pi k} \epsilon^{-\gamma} \quad (16)$$

It can be shown that  $f(z)$  is monotonically increasing, from which we conclude that there is an increase in  $n_{opt}$  whenever the right hand side of (16) increases. Thus, an increasing number of processors,  $P = p^d$ , or increasing communication cost,  $\beta \frac{B}{A}$ , leads to an increase in the optimal order of approximation, and corresponding decrease in the optimal number of subdomains, subject to the constraint that both be integers.

## 5 Results

In this section, we examine the parameters which influence  $(q_{opt}, n_{opt})$  in a multi-processor implementation. Both two- and three-dimensional problems are considered,

Table 1: Problem, algorithm, and hardware parameters.

d	k	$e^{-\epsilon}$	$\gamma$	A	B	$\alpha$	$\beta$	$t_a$
2,3	16,32	$10^{-2}, 10^{-6}$	0.75	$4d^2$	$4d$	1000	10	$10^{-8}s$



with maximum resolvable wave numbers of  $k = 32$ , and 16, respectively. These, along with the specified error tolerance, specify the characteristics of the physical problem. The algorithmic characteristics have been selected to reflect a matrix-matrix product based approach to differentiation, i.e., with a fairly low constant  $A$  and relatively high exponent  $\gamma$ . The multiplier  $d$  in the communication term  $B$  reflects the fact that information is propagated in each of  $d$  spatial dimensions whenever the linear convection operator is applied. The hardware parameters are representative of dedicated parallel architectures. The nondimensional communication characteristics  $\alpha$  and  $\beta$  are derived from tests described in Appendix A. For networks of workstations, the latency term,  $\alpha$ , would generally be much higher. While this impacts the overall parallel efficiency, it would not impact the optimal discretization, as noted earlier. All the results scale directly with the floating point cycle time  $t_a$ . However, to give the times a realistic scale,  $t_a$  is set to  $10^{-8}$  seconds, corresponding to a nominal performance of 100 MFLOPS per node. Table 1 summarizes the baseline parameters.

The impact of communication overhead is illustrated in Table 2. Values of  $(q_{opt}, n_{opt})$  computed from (15) and (10), along with model estimates of time per step,  $T_P$ , are shown for  $P = 1$  to 4096. In order to highlight the slight variations in  $(q_{opt}, n_{opt})$  the values have not been rounded up to the nearest integer. In addition, we have momentarily relaxed the constraint  $q = mp$ ; the impact of granularity is examined below.

The results of Table 2 indicate that communication overhead  $T_\beta$  has strikingly little effect on the value of  $q_{opt}$ . The case ( $d = 2$ ,  $e^{-\epsilon} = 10^{-2}$ ) is the only one which exhibits notable (20 %) variation in  $q_{opt}$ . Even if the communication/computation ratio is artificially increased by halving  $\gamma$ , or doubling  $\beta$ , the results are little changed. Note that if it is necessary to communicate data on the edges in addition to the faces, an approximate bound on the number of words transferred is  $36 \frac{qn}{p}$  in  $\mathbb{R}^3$  (twelve edges of length  $\frac{qn}{p}$  to three processors each). For optimal values of  $(q, n)$ , this is significantly below the amount of face data transferred ( $6 \left(\frac{qn}{p}\right)^2$ ) and we conclude that the edge data has little influence on the optimal discretization. Finally, note that increasing the physical complexity by increasing  $k$  simply causes a proportional increase in  $q_{opt}$ , and no change in  $n_{opt}$ .

It is useful to examine the sensitivity of solution time to the choice of  $(q, n)$ . Fig. 1 shows the normalized time-per-step,  $PT_P$ , versus  $n$  for error criteria  $10^{-3}$ ,  $10^{-6}$ , and  $10^{-9}$ , with  $P = 1, 4, 16, 64$ , and 256. These results were computed using (9) in conjunction with (10). In this case,  $\alpha$  was set to zero, simply to allow the multi-processor results to collapse onto a single curve, and to highlight the influence of the remaining communication term. Also shown are the values of  $PT_P$  when  $q$  is restricted to be an integer, i.e.,

$$q = \left\lceil \frac{e\pi k}{2n} e^{\frac{\epsilon}{n}} \right\rceil. \quad (17)$$

These are seen in Fig. 1 as the jagged counterparts to the smooth curves derived directly from (9-10). The fact that the multi-processor curves lie on top of one another implies near unity efficiency, i.e.,  $t_1/(P t_P) \approx 1$ , which would be the case if the latency were in fact zero. The most striking feature of Fig. 1 is that the continuous  $(q, n)|_\epsilon$  curves exhibit a broad minimum, especially for smaller error tolerance,  $error =$

$e^{-\epsilon}$ . By contrast, the time for the *discrete*  $(q, n)$  pairs exhibits large fluctuations, particularly as  $q$  nears unity. To exploit the broad minima of the continuous results, it is best to choose  $q > q_{opt}$ , with appropriate  $n$ , in order to minimize deviation from the optimal solution times.

The results presented so far appear to indicate that parallelism has little influence on the discretization pair  $(q, n)$ . In fact, a fundamental constraint which has not been imposed in the previous model problems is that the number of subdomains,  $Q$ , be an integral multiple of the number of processors,  $P$ . The combined influences of this imposed granularity, nonzero latency, and suboptimal  $(q, n)$  pairs are shown in Fig. 2. The solution time versus number of processors is plotted for the two-dimensional baseline case (Table 1) with error criteria  $e^{-\epsilon} = 10^{-3}$  and  $10^{-6}$ . Four values of  $n$  are considered: a fixed (arbitrary) value of  $n = 4$ , the optimal serial case,  $n = n_{opt_s}$ , the optimal parallel case,  $n = n_{opt}$ , and the value of  $n$  obtained if  $q = p$ , i.e., corresponding to one subdomain per processor.

For this problem, the low-order ( $n = 4$ ) case requires significantly more time than the other cases. The optimal curves, which are essentially indistinguishable, exhibit significant deviation from linear speed up for  $P > 256$ , due to latency. However, for  $P > 64$ , the optimal number of subdomains,  $q_{opt}$ , is less than  $p$ , implying that the analytically derived optimal performance cannot be obtained in this regime. Instead, the number of subdomains must be set equal to  $P$ , and the performance must track the  $q = p > q_{opt}$  curve. Thus, imposed granularity is an additional contributor to inefficiency. We note that, in practice, this source of inefficiency might not be encountered since it is common to increase the number of processors with increasing physical complexity and problem size, resulting in “scaled” speed-up, as noted in [8]. Since  $q_{opt}$  scales almost directly with physical complexity ( $k$ ), it should be possible to maintain  $q \approx q_{opt}$  in the scaled speed-up case.

Table 2: Optimal discretization pairs and time-per-step (s)

				$e^{-\epsilon} = 10^{-2}$			$e^{-\epsilon} = 10^{-6}$		
$d$	$k$	$P$	$p$	$q_{opt}$	$n_{opt}$	$T_P$	$q_{opt}$	$n_{opt}$	$T_P$
2	32	1	1	11.9	12.3	$2.2 \times 10^{-2}$	4.0	36.8	$5.1 \times 10^{-2}$
2	32	4	2	11.8	12.3	$5.7 \times 10^{-3}$	4.0	36.9	$1.3 \times 10^{-2}$
2	32	16	4	11.7	12.4	$1.5 \times 10^{-3}$	3.9	37.0	$3.3 \times 10^{-3}$
2	32	64	8	11.6	12.5	$4.1 \times 10^{-4}$	3.9	37.2	$8.5 \times 10^{-4}$
2	32	256	16	11.3	12.8	$1.3 \times 10^{-4}$	3.9	37.5	$2.5 \times 10^{-4}$
2	32	1024	32	10.7	13.3	$6.6 \times 10^{-5}$	3.8	38.2	$9.4 \times 10^{-5}$
2	32	4096	64	9.8	14.2	$4.7 \times 10^{-5}$	3.6	39.5	$5.4 \times 10^{-5}$
3	16	1	1	3.5	18.4	$8.6 \times 10^{-1}$	1.2	55.3	2.0
3	16	8	2	3.5	18.6	$1.1 \times 10^{-1}$	1.2	55.4	$2.5 \times 10^{-1}$
3	16	64	4	3.4	18.7	$1.4 \times 10^{-2}$	1.2	55.6	$3.1 \times 10^{-2}$
3	16	512	8	3.4	19.0	$1.8 \times 10^{-3}$	1.1	56.0	$4.0 \times 10^{-3}$
3	16	4096	16	3.3	19.5	$2.9 \times 10^{-4}$	1.1	56.7	$5.6 \times 10^{-4}$

## 6 Conclusions

Analytical error estimates derived in [1] have been used in conjunction with computational complexity estimates for parallel spectral subdomain algorithms to derive optimal discretization parameters for time-explicit numerical solution of hyperbolic partial differential equations. Communication overhead increases the optimal approximation order,  $n_{opt}$ , over that derived for the serial case. However, for realistic multicomputer parameters, the change is typically small. Moreover, because the optimal solution time exhibits a broad minimum about  $n_{opt}$ , while the achievable solution time (due to integral constraints on  $q$  and  $n$ ) exhibits large amplitude fluctuations for large  $n$ , there is incentive to choose  $n < n_{opt}$ . The most significant impact of parallelism upon the choice of  $(q, n)$  is that it may potentially impose a granularity upon the discretization which is suboptimal, i.e,  $q = p > q_{opt}$ .

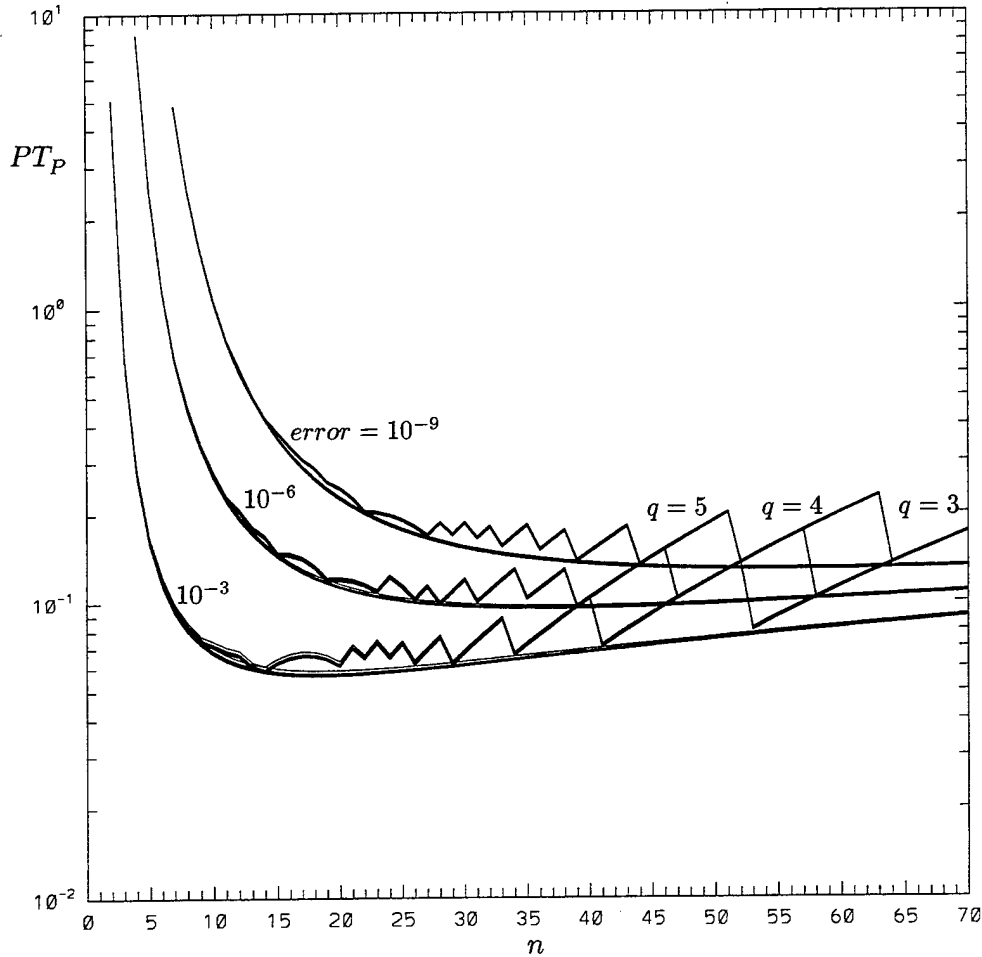


Figure 1: Normalized time per step for two-dimensional problem with  $\alpha = 0$ .

## Appendix A: Communication parameters

We assume that contention free messages consisting of  $m$  words have a point-to-point travel time given by:

$$t_{comm}(m) = (\alpha + \beta m)t_a ,$$

The model does not account for distance between processors. However, for most modern message passing systems, any fluctuation in latency due to greater wire separation between processors is so overwhelmed by other sources of latency as to be immeasurable.

The values of  $\alpha$  and  $\beta$  are measured via the following point-to-point or “ping-pong” test. For each processor pair,  $(0, k)$ ,  $k = 2^i - 1$ ,  $i = 1, \dots, \log_2 P$ , the processors synchronize and commence timing. Processor 0 then sends an  $m$ -word message to processor  $k$ , and immediately posts a blocking receive. Processor  $k$  posts a blocking receive, and upon receipt of the message from node 0, sends an  $m$ -word segment of a different array back to 0. This continues for fifty iterations, each message sent from and placed into different memory locations in order to avoid unduly favorable cache behavior. Fig. A shows the measured round-trip message times,  $t_{rt} = 2(\alpha + \beta m)t_a$ , for the 512-node Intel Delta at Caltech, the 340 node Intel Paragon at Wright-Patterson AFB, and the 20-node IBM SP2 at Brown University using the `mpich` message passing library from Argonne National Labs.

Estimates of  $t_a$  are derived from MFLOPS measurements for computation/memory-access patterns which are typical of the computations under consideration. In this case, the work is dominated by matrix-matrix products of order  $n$ , which have favorable caching patterns but generally do not attain near peak-performance unless the matrix order is quite large, i.e.,  $n \approx 30$  or greater.

The measured values of  $t_a$ ,  $\alpha t_a$  and  $\beta t_a$  are shown in Table A, along with derived

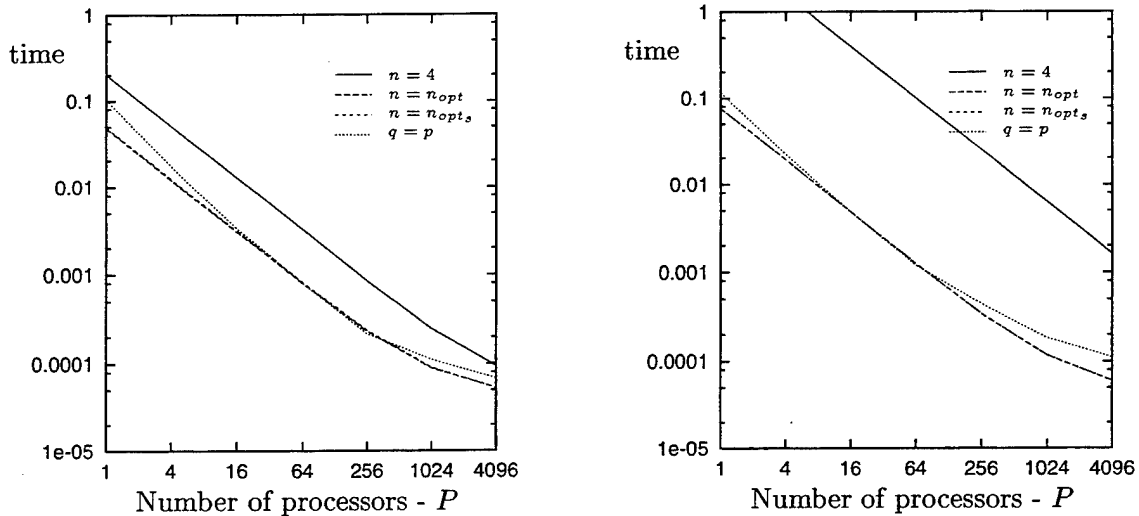


Figure 2: Time per step for two-dimensional problem with truncation error of  $e^{-\epsilon} = 10^{-3}$  (left) and  $10^{-6}$  (right).

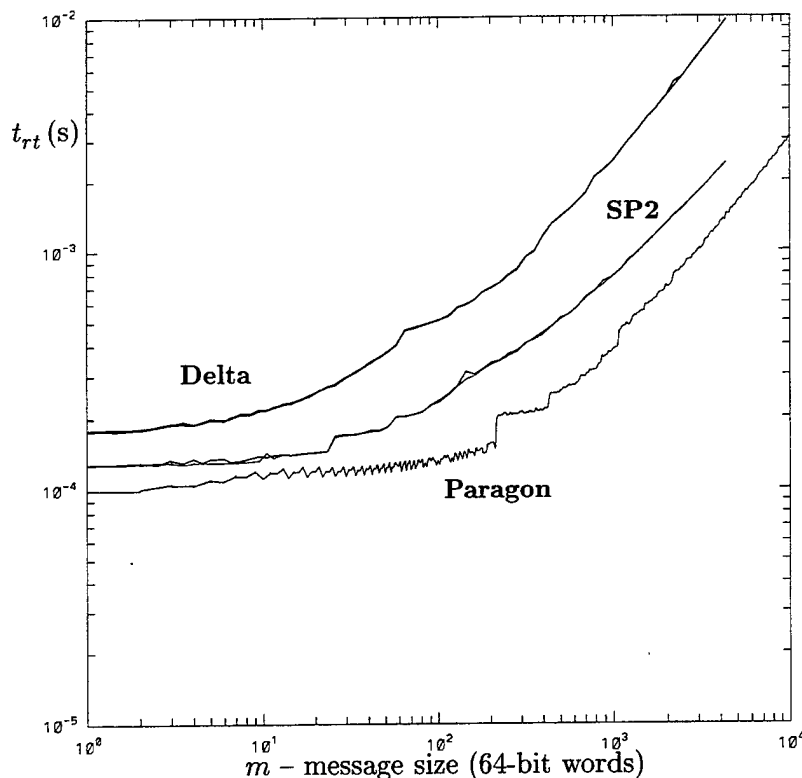


Figure A: Measured round-trip message times,  $t_{rt}$ , for ping-pong test.

estimates for  $\alpha$  and  $\beta$ . Also listed is  $m_2 = \alpha/\beta$ , the characteristic message size distinguishing short (latency-dominated) from long (bandwidth-limited) messages.

Table A: Machine dependent parameters 64-bit arithmetic

Machine	$t_a$ (s)	$\alpha t_a$ (s)	$\beta t_a$ (s)	$\alpha$	$\beta$	$m_2$
Delta	$1.0 \times 10^{-7}$	$9. \times 10^{-5}$	$1.1 \times 10^{-6}$	900	11	80
Paragon	$.66 \times 10^{-7}$	$5. \times 10^{-5}$	$.15 \times 10^{-6}$	760	2.3	330
SP2	$.20 \times 10^{-7}$	$6. \times 10^{-5}$	$.27 \times 10^{-6}$	3000	15	200

## References

- [1] Gottlieb, D.I. and Wasberg, C.E. 1996. "Optimal strategy in domain decomposition spectral methods for wave-like phenomena," in *Proceedings of the 9th Int. Conf. on Domain Decomposition Methods*, Bergen, Norway, AMS.
- [2] Keller, H.B. and Schreiber, R. 1983. "Driven cavity flows by efficient numerical techniques." *J. Comp. Phys.* 49(2):310-333.
- [3] Chan, T. and Shao, J.P. 1994. "Optimal coarse grid size in domain decomposition." *J. Comp. Math.* 12(4):291-297.
- [4] Rønquist, E.M. 1988. Ph.D. Thesis, Dept. of Mech. Eng., Massachusetts Institute of Technology.

- [5] Fischer, P.F. and Patera, A.T. 1991. "Parallel Spectral Element Solution of the Stokes Problem." *J. Comput. Phys.* 92:380-421.
- [6] Gear, C.W. 1971. *Numerical initial value problems in ordinary differential equations*, Englewood Cliffs, NJ, Prentice-Hall.
- [7] Carpenter, M., Gottlieb, D.I., and Abarbanel, S. 1995. "Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes" *J. Comp. Phys.* (to appear).
- [8] Gustafson, J.L., Montry, G.R., and Benner, R.E. 1988. "Development of Parallel Methods for a 1024-Processor Hypercube," *SIAM J. on Sci. and Stat. Comput.* 9(4):609-638.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1996	3. REPORT TYPE AND DATES COVERED Contractor Report		
4. TITLE AND SUBTITLE ON THE OPTIMAL NUMBER OF SUBDOMAINS FOR HYPERBOLIC PROBLEMS ON PARALLEL COMPUTERS		5. FUNDING NUMBERS  C NAS1-19480 WU 505-90-52-01		
6. AUTHOR(S) Paul Fischer David Gottlieb				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Computer Applications in Science and Engineering Mail Stop 403, NASA Langley Research Center Hampton, VA 23681-0001		8. PERFORMING ORGANIZATION REPORT NUMBER  ICASE Report No. 96-68		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CR-201625 ICASE Report No. 96-68		
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Dennis M. Bushnell Final Report To appear in the International Journal of Supercomputing.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Unclassified-Unlimited  Subject Category 60, 61		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) The computational complexity for parallel implementation of multidomain spectral methods is studied to derive the optimal number of subdomains, $q$ , and spectral order, $n$ , for numerical solution of hyperbolic problems. The complexity analysis is based upon theoretical results which predict error as a function of $(q, n)$ for problems having wave-like solutions. These are combined with a linear communication cost model to study the impact of communication overhead and imposed granularity on the optimal choice of $(q, n)$ as a function of the number of processors. It is shown that, for present day multicomputers, the impact of communication overhead does not significantly shift $(q, n)$ from the optimal uni-processor values, and that the effects of granularity are more important.				
14. SUBJECT TERMS Domain Decomposition; Spectral Methods; Parallel Computing			15. NUMBER OF PAGES 14	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	